



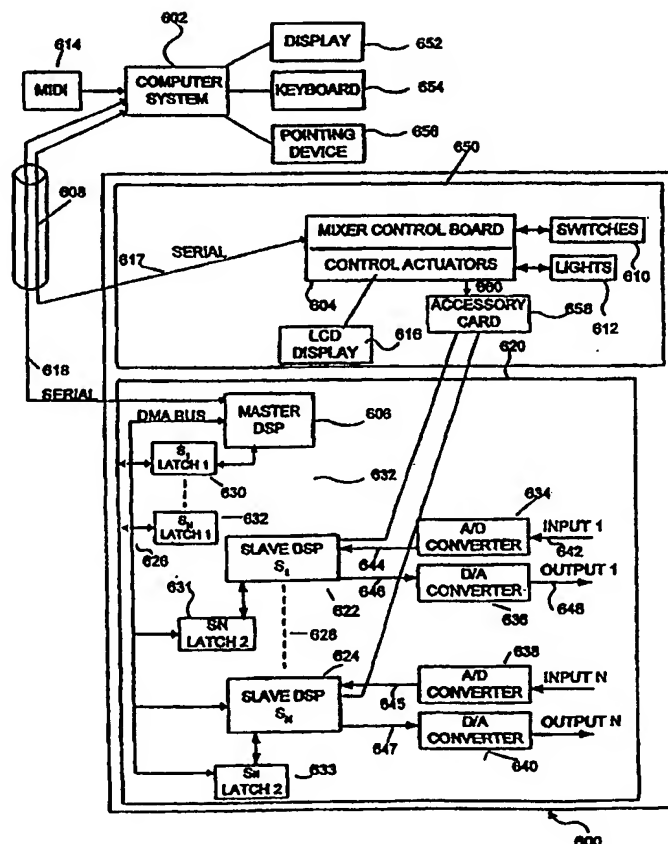
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : <b>H04B 1/00</b>	<b>A1</b>	(11) International Publication Number: <b>WO 99/37032</b> (43) International Publication Date: 22 July 1999 (22.07.99)
<p>(21) International Application Number: PCT/US98/16590</p> <p>(22) International Filing Date: 10 August 1998 (10.08.98)</p> <p>(30) Priority Data: 09/007,456 15 January 1998 (15.01.98) US</p> <p>(71) Applicant: MACKIE DESIGNS INC. [US/US]; 16220 Wood-Red Road N.E., Woodinville, WA 98072 (US).</p> <p>(72) Inventors: MACKIE, Greg, C.; 8307 N.E. Juanita Drive, Kirkland, WA 98034 (US). WATTS, Peter, F.; 10318 N.E. 201st Place, Bothell, WA 98011 (US). TUDOR, Robert, J.; 6509 - 57th Avenue S.E., Snohomish, WA 98290 (US).</p> <p>(74) Agent: BERGSTROM, Robert, W.; Weiss Jensen Ellis &amp; Howard, Suite 2600, 520 Pike Street, Seattle, WA 98101 (US).</p>		<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published With international search report.</p>

(54) Title: DIGITAL SIGNAL MIXING ARCHITECTURE

## (57) Abstract

An improved digital signal mixing architecture for applications such as digital mixing consoles is provided. The architecture provides a software configurable digital mixing system that is expandable to more than 48 channels. The architecture comprises a master DSP device (606) coupled to a number of slave DSP devices (622, 624) via a DMA bus (DMA BUS). The master DSP performs functions including data bus arbitration, calculating coefficient variables, and transferring the variables to slave DSP's. The master DSP is interconnected to each of the slave DSP's to write signal processing results to the internal memories of each of the other slave DSP's. Once each slave DSP has the results of every processed input signal, the master DSP commands combinations of the slave DSP's to mix the processed input signals to create output signals (646, 647). Mixing combinations are controlled by scripts that are interpreted by software. The digital mixing architecture of the present invention provides a system that is software configurable, so that a mixing console embodying the invention is easily re-configurable and upgradeable.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## DIGITAL SIGNAL MIXING ARCHITECTURE

### Related Applications

This application is a continuation-in-part application of co-pending  
5 U.S. application Serial No. 09/007,456 filed January 15, 1998, which is a  
continuation-in-part of U.S. provisional application Serial No. 60/035,520, filed  
January 15, 1997.

### Field Of The Invention

10 The present invention relates generally to digital signal processing, and more  
particularly, to an improved architecture for a digital signal mixer for processing  
audio signals.

### Background Of The Invention

An audio mixer serves as the central component of a professional audio  
15 system by electronically blending, routing, and enhancing sound sources, such as  
voices, musical instruments, sound effects, audio tape, video tape, and other live or  
pre-recorded material. Audio mixers are commonly analog devices that mix audio  
signals. Analog signals are used both because analog technologies were available  
prior to the more recent digital technologies, and because inputs to audio mixers and  
20 outputs from audio mixers are commonly analog signals. However, analog mixers  
have numerous limitations in comparison to digital mixers, including a limited ability  
to recall or reconfigure the signal flow. Therefore, the professional audio mixer  
market (estimated by industry sources at approximately \$1,000,000,000 in 1998) is

moving from audio mixers based on analog signal processing to audio mixers based on digital signal processing. Digital signals can be memorized and recalled, unlike analog signals. The console parameters of a digital audio mixer can be reconstructed from memorized digital signals, allowing a digital audio mixer to provide a total recall of all console parameters. Digital audio mixers can therefore provide play back of recalled digital signals and console parameters by a process called "real-time automation." Digital signals can be easily transported to remote processors via the digital data transport facilities developed and used in computer networks. Digital signals are easily sent through digital data processing interfaces, including interfaces to communications devices and data storage devices. A wealth of possible enhancements and transformations of digital signals are available, as well. But, to date, digital signal mixers are very expensive, with prices varying up to \$1,000,000 per unit for digital mixers that, because of their size, are useful only in a single location. A need has therefore been recognized in the industry for a digital audio mixer that costs less than \$15,000 per unit, that is small enough to be portable, and that provides a multiplicity of functions, including real time processing, a control processing unit with real time upgradeable and reconfigurable capabilities, remote processing and manipulation through device interfaces, hard disk and floppy drives, on-board and remote video monitors, and audio effects integration.

## Summary Of The Invention

One embodiment of the present invention is a digital signal mixing architecture for digital audio mixers that can provide from 1 to a very large number of input and output channels, limited only by the bandwidths and capacities of various internal components, including a high-speed data bus. A preferred embodiment provides a digital audio mixer capable of providing from 1 to 48 input and output channels. A number of these digital audio mixers may be combined to form audio mixing systems capable of mixing more than 48 channels.

In a preferred embodiment, the digital signal mixing architecture comprises a number of integrated digital signal processors ("DSPs"), including a master DSP

coupled to a number of slave DSPs on a direct memory access ("DMA") bus. The master DSP performs functions that include data bus arbitration, calculating coefficient variables that specify output signals for each slave DSP, and transferring the coefficient variables to the slave DSPs.

5        Each slave DSP is preferably coupled to a number of audio converters. An audio converter converts an input analog signal to a digital signal for processing by the slave DSP and converts a processed digital signal to an analog signal for output. A channel is a single input signal paired with a single output signal, and each channel is commonly associated with an input converter and an output converter. Each slave  
10    DSP can process a number of channels, typically ranging from 1 channel to 8 channels, that depends upon such factors as the number of audio converters, the processing power requirements, and the system functionality, such as equalization and dynamics processing. In different embodiments, a slave DSP may process more than 8 channels. A slave DSP may also be coupled directly to digital input signals  
15    and digital output signals. Audio converters are not needed for such digital input and output channels. In different embodiments, a selected channel may thus provide analog-to-digital ("A/D") input conversion and digital-to-analog ("D/A") output conversion, A/D input conversion and digital output, digital input and D/A output conversion, or digital input and digital output.

20        For example, a 48-channel digital audio mixer designed according to an embodiment of the digital signal mixing architecture of the present invention may comprise a master DSP and 24 slave DSPs. If each slave DSP process has two input channels and two output channels, the resulting audio mixing system has 48 inputs and 48 outputs. As another example, a 16-channel digital audio mixer designed  
25    according to an embodiment of this digital signal mixing architecture may comprise a master DSP coupled to 4 slave DSPs, with each slave processing 4 input/output channels of audio signals. This configuration results in an audio mixing system having 16 inputs and 16 outputs. An audio mixing system having greater than 48 channels can be achieved by providing a top-level master DSP coupled to a number of

second-level master DSPs, each second-level master DSP connected to a number of slave DSPs. Such a combination of two 48-channel audio mixing systems, each comprising a second-level master DSP coupled to 24 slave DSPs, results in an audio mixing system capable of mixing 96 channels.

5           The master DSP is coupled to each of the slave DSPs, and the slave DSPs are interconnected in a manner that enables the master DSP to command each of the slave DSPs to write the results of the slave DSP's signal processing to the internal memory of each of the other slave DSPs. For example, if there are 4 slave DSPs that each process two input channels to produce two processed signals, each slave DSP will  
10   have 8 processed input signals stored in the slave DSP's internal memory. Once each slave DSP has the results of every processed input signal in the slave DSP's internal memory, the master DSP can command any combination of the DSP slaves to mix any of the processed input signals to create a mixed digital output signal that is sent to one or more channels. The mixed digital output may be first directed through an  
15   audio converter, for D/A conversion, in order to output an analog signal corresponding to the mixed digital output.

          The master DSP device is initially activated to prepare the master DSP to receive data from a computer system coupled via a data cable to the digital audio mixer. The computer system downloads a set of instructions, known as an executable  
20   file, to the master DSP. Once the master DSP has received the executable file from the computer system, the master DSP is reinitialized to process the executable file. The computer system then downloads executable slave DSP files which the master DSP distributes via the DMA data bus to the slave DSPs, each of which then writes the executable slave DSP file received from the master DSP into the slave DSP's  
25   internal memory. The slave DSPs then initialize themselves to prepare the digital audio mixer for running a digital audio signal mixing sequence specified by the downloaded executable files.

          Analog input signals are first converted to representative digital samples by the audio converters. A stream of digital samples is then transmitted to the inputs of

the slave DSPs for processing. The slave DSPs perform various signal enhancement functions, such as equalization and compression, on each digital sample and place the resulting enhanced digital samples into a selected output memory location. After each input signal has been processed to create an enhanced digital sample by the slave DSP  
5 associated with the input signal, the master DSP instructs a first slave DSP to place the first slave DSP's digital samples on the DMA data bus. The master DSP may then latch one or more of the first slave DSP's digital samples into the memory of other selected slave DSPs according to instructions contained in the master DSP's executable file. The master DSP then instructs a second slave DSP to place the  
10 second slave DSP's digital sample on the DMA data bus. The master DSP may latch one or more of the digital samples from the second slave DSP into the memory of other selected slave DSPs. This process is repeated until the digital samples from each slave DSP have been placed into the memory of all other slave DSPs into which each slave DSP's digital samples were instructed to be placed.

15 Once digital samples from each of the slave DSPs has been correctly routed according to the instructions, the master DSP generates a signal that instructs the slave DSPs to mix the digital samples in various combinations to create a number of digital output signals. The various combinations of mix are determined by default coefficient variables contained in the slave DSPs' executable files that are loaded into  
20 the memory of the slave DSPs when the slave DSPs' executable files are initially downloaded. Each slave then outputs the mixed digital output signal generated by the slave DSP, as instructed.

This process is continuously repeated, and is only interrupted when the master DSP receives a new instruction from the computer system. A new instruction may be  
25 generated, for example, when a user alters the settings of the digital audio mixer by manipulating controls on the console of the digital audio mixer. When the new instruction is received from the computer system, the master DSP calculates new coefficient values, places the new values onto the DMA bus, and writes the values

into the memory of the selected slave DSPs. The new coefficient values overwrite previously written, or default values, and specify subsequent audio outputs.

Scripts are employed by the computer system to configure and control the digital audio mixer. An image of a digital audio mixer console, i.e., a virtual console, may be displayed on a video monitor and the user may employ a keyboard and/or a pointing device to manipulate the virtual console. Scripts are written in plain text. The computer system provides an editor that may be employed to edit the scripts. By editing the scripts, a user may enable and disable features of the virtual console and customize the virtual console. Script updates may be downloaded from the computer system. Point to point modems, network interface cards, and hubs may be employed to connect the computer system to remote computer systems. New scripts and updates are data encrypted using the serial number of the digital audio mixer as the public encryption key. In a preferred embodiment, Riveat, Shamin, Adleman ("RSA") data encryption is used. However, other data encryption methods may be employed to provide data security. Additionally, a look-up table may be employed to produce text displayed to the user in any one of a number of languages that may be selected by a user.

Thus, one embodiment of the digital mixing architecture of the present invention specifies a digital audio mixer that is software-configurable and, therefore, is easily reconfigurable and upgradeable. Software-configurability and channel expandability provides a digital mixing console that is low cost to upgrade or reconfigure.

### **Brief Description Of The Drawings**

FIGURE 1 illustrates the basic signal path within a digital audio mixer.

FIGURE 2 is a schematic illustration of the processing of a number of different input analog signals by a digital audio mixer.

FIGURE 3 shows the path of an input signal to a slave DSP.

FIGURE 4 shows the processing of an input signal by a DSP logic circuit.



FIGURE 5 shows completion of a processing and mixing operation by a slave DSP.

FIGURE 6 shows an overview of the hardware architecture of a digital audio mixer designed according to one embodiment of the present invention.

5       FIGURE 7 is an overview flowchart of the logical steps performed by the DSP board in response to receiving instructions from the computer system.

FIGURE 8 shows an exemplary computer system incorporated in one embodiment of a digital audio mixer.

10       FIGURE 9 shows an overview of the software employed by the present invention.

FIGURE 10 shows an overview of an automatic training sequence executed by the automation engine implemented in accordance with one embodiment of the present invention.

15       FIGURE 11 shows an overview of the startup sequence executed by the automation engine to initialize the digital audio mixer.

FIGURE 12 shows further details of configuration script interpretation by the automation engine.

FIGURES 13A, 13B, and 13C show an exemplary desktop script employed by the mixer script engine.

20

### **Description Of The Invention**

25       FIGURES 1 through 5, and the discussion of those Figures below, provide an overview of one embodiment of the digital signal mixing architecture of the present invention. FIGURE 1 illustrates the basic signal path within a digital audio mixer. A digital audio mixer commonly receives a number of analog input signals like, for example, the analog input signal 102 shown in FIGURE 1. The digital audio mixer processes the received analog input signals and outputs a number of analog output signals that result from the processing, for example, the analog output signal 104 shown in FIGURE 1. The analog input signal, signal 102 is passed through an analog-to-digital converter ("ADC") 130 to produce a digital input signal 132 that is

input to the slave DSP 134. The slave DSP 134 outputs a digital output signal 136 which is passed through a digital-to-analog converter ("DAC") 138 to produce the final analog output signal 104.

5 A digital audio mixer used for coordinating a live musical performance may, for example, input analog signals from a number of microphones, from electrical instrument pickup devices, and from electronic instruments, such as synthesizers, process the input signals in various ways, and then output analog signals to a number of amplifiers and public address systems that broadcast the music to the audience. The processing performed by the digital audio mixer may include equalization of the  
10 input signals over the audio frequency range, compression of the input signals, altering the volume, or amplitude, of the input signals, blending of multiple input signals into a single output signal, and extraction of multiple output signals from a single input signal.

One embodiment of the present invention provides for processing of input  
15 signals by a number of slave DSPs 106, 108, 110, and 112. Each slave DSP is responsible for processing the input signals provided by a number of channels. For example, slave DSP 106 is shown in FIGURE 1 receiving input signals from four inputs 114, 116, 118, and 120, and outputting signals to four outputs 122, 124, 126, and 128. Thus, in the expanded view 106 of slave DSP 134, below, input  
20 signals 114, 116, 118, and 120 have each been processed by a separate ADC and are therefore digital input signals. Likewise, digital output signals 122, 124, 126, and 128 are digital signals that will be subsequently passed through separate DACs to produce final analog output signals.

In some embodiments, certain channels of the digital audio mixer may have  
25 digital input, digital output, or both digital input and output. Such channels may be used by instruments, such as digital synthesizers, that produce digital output directly, without requiring analog-to-digital ("A/D") conversion. Likewise, a digital output channel might be directed to an electronic device that can directly process digital signals, without requiring digital-to-analog ("D/A") conversion.

The separate digital input signals 114, 116, 118, and 120 are processed by a DSP logic circuit 140 within the slave DSP 106 to produce the output signals. Output signals corresponding to each of the input signals are stored in memory locations 142, 144, 146, and 148 within a memory device 149 included in the slave DSP 106. In addition, the memory device 149 within the slave DSP 106 also stores output signals from other channels processed by the other slave DSPs 108, 110, and 112. These additional memory locations are indicated in FIGURE 1 by dotted line 150 and the final memory location 152. The slave DSP then combines one or more of the output signals stored in the memory locations by means of mixing logic circuit 154 to produce final digital output signals 122, 124, 126, and 128. Thus, a single digital output signal for a particular channel, such as digital output signal 122 for channel 1, may include signal components from any of the input signals of any channel processed by any of the slave DSPs within a digital audio mixer.

FIGURE 2 is a schematic illustration of the processing of a number of different input analog signals by a digital audio mixer. Analog input signals 201-208 are converted to digital input signals by ADCs 209-216. The digital input signals 217-229 are processed by slave DSPs 225-232 to produce digital output signals 233-240. The digital output signals 233-240 are converted to analog output signals 249-256 by DACs 241-248. Again, as discussed above, slave DSPs 225-232 process the input analog signals, including equalizing and compressing the input signals, and combine various combinations of the processed input signals to produce each digital output signal. Also, as discussed above, certain of the input signals may be input in digital, rather than in analog, form and therefore need not be converted by an ADC, and certain of the output signals may be output in digital form rather than in analog form and thus do not require conversion by a DAC.

A master DSP 258 controls the operations and communication of the slave DSPs. The master DSP 258 and the slave DSPs 225-232 exchange data and commands via a DMA bus 260. The master DSP is responsible for DMA bus arbitration. The master DSP transfers coefficient variables via the DMA bus 260 to

each of the slave DSPs 225-232. The transferred coefficient variables specify the processing and mixing operations to be performed by each slave DSP. Each slave DSP receives a separate set of coefficient variables. The master DSP receives the coefficient variables from a computer system 262 via a serial bus 264. The computer system 262 downloads an executable file (not shown) via the serial bus 264 to the master DSP 258. The master DSP is then reinitialized and executes the downloaded executable file. The computer system 262 then downloads, via serial bus 264 to the master DSP, executable slave DSP files (not shown) which the master DSP 258 distributes via the DMA bus 260 to each of the slave DSPs 225-232. These slave DSP executable files contain default coefficient variables and other instructions to control the slave DSPs. Thus, the input and output of the audio digital mixer is controlled via instructions included in executable files transferred from the computer system 262 via serial bus 264 to the master DSP 258 and, via the master DSP 258 and DMA bus 260, to the slave DSPs 225-232.

FIGURES 3-5 illustrate, in greater detail, the processing and mixing operations performed within the digital audio mixer. Many of the components shown in FIGURE 3 are equivalent to components shown in FIGURES 1 and 2. In the interest of brevity, these components in FIGURES 3-5 will be labeled with the numerical labels with which they were labeled in FIGURES 1 and 2. The above descriptions of these components will not be repeated.

FIGURE 3 shows the path on an input signal to a slave DSP. It is assumed in FIGURE 3 that the executable instruction files that control operation of the master DSP 258 and slave DSP 106 have been downloaded from the computer system 262 and that the master DSP 258 and slave DSP 106 have been properly initialized and are ready to perform the processing and mixing operations. The analog input signal 102 passes through the ADC 130 and is input as a digital input signal 114, corresponding to channel 1, to the slave DSP 106. The slave DSP 106, under control of the master DSP 258, essentially continuously performs processing and mixing operations on the input signal 114, as well as input signals 116, 118, and 120, and

perhaps additional digital outputs from other slave DSPs, to continuously produce digital outputs on digital output signal lines 122, 124, 126, and 128. In essence, the slave DSP 106 performs a processing and mixing operation at regular, short time intervals like, for example, every 10 to 20 microseconds. FIGURES 4 and 5 below  
5 illustrate a single processing and mixing operation.

In FIGURE 4, input signal 114 is processed by the DSP logic circuit 140 and the resulting processed signal is stored in memory location 142, corresponding to channel 1. In the processing and mixing operation, the slave DSP 106 will also process input signals 116, 118, and 120, and places the resulting processed signals in  
10 memory locations 144, 146, and 148, respectively. The master DSP 258 then cycles through each of the slave DSPs, including slave DSP 106, and instructs each slave DSP to make that slave DSP's processed signals available to the DMA bus 260 for transfer to memory locations within other slave DSPs. The master DSP 258 then controls the DMA bus 260 to transfer certain of the processed signals from each slave  
15 DSP to memory locations in other slave DSPs, according to instructions that the master DSP 258 received from the computer system 262 in the executable instruction file. Each signal to be transferred is moved from the memory of a slave DSP into a latch in one operation and, in a second operation, is transferred from the latch via the DMA bus 260 to other slave DSPs.

FIGURE 5 shows completion of a processing and mixing operation by a slave DSP. After the processed signals have been stored in the memory locations of each slave DSP, and the processed signals have been exchanged between slave DSPs in order that each slave DSP has, in its memory, all the processed signals that the slave DSP will need for the mixing operation, the master DSP 258 instructs the slave  
20 DSP 106 to mix, in the mixing logic circuit 154, the processed signals stored in memory to produce output signals according to the executable instruction files downloaded from the computer system 262 via the serial bus 264, master DSP 258, and DMA bus 260.  
25

The mixing logic circuit 154 of the slave DSP 106 executes instructions from the downloaded executable instruction file that directs the mixing logic circuit 154 to combine one or more processed signals stored in memory in order to create each output digital signal. For example, output signal 122 of channel 1 may include a mix of processed signals stored in memory locations corresponding to channel 2 144, channel 3 146, and perhaps another channel 502 for which the processed signal is computed by a different slave DSP. The slave DSP 106 produces digital output signals for all channels that the slave DSP is directed to produce output digital signals for by instructions included in the downloaded executable instruction file, and sends the digital output signals through a DAC, for example DAC 138, to produce a final analog output signal, such as analog output signal 104. The signal processing and mixing operation is repeated at fixed time intervals like, for example, every 20 microseconds, in order to produce a sufficient number of digital output signals per second that, after D/A conversion, produces the perception in a human listener of a continuous analog output signal.

When a user of the digital audio mixer manipulates the console of the digital audio mixer in order to change the output signal for one or more channels, the console manipulations are interpreted by the computer system 262. The computer system 262 then produces executable instruction files that reflect the new settings of the console of the digital audio mixer and downloads these new executable instruction files to the master DSP 258 and slave DSPs 225-232, as discussed above. The master DSP 258 and slave DSPs 225-232 are then reinitialized and begin processing, mixing, and outputting signals in accordance with the new set of instructions. The download of the executable instruction files and the reinitialization of the DSPs is interleaved with the ongoing input/output signal processing being conducted by the digital audio mixer so that no perceptible discontinuities are produced in the output audio signal. Thus, for example, a user may alter the volume or equalization of one or more channels by continuously moving knobs or dials on the console of the digital audio

mixer to produce the perception of a continuously changing audio output signal that varies corresponding to manipulation of the volume and equalization controls.

FIGURE 6 shows an overview of the hardware architecture of a digital audio mixer designed according to one embodiment of the present invention. A digital audio mixer 600 includes a computer system 602 that is coupled to a digital audio mixer control board 604 and to a master DSP 606 via one or more communication links. FIG. 6, for example, shows the computer system 602 coupled to the digital audio mixer control board 604 and the master DSP 606 via two serial channels 617 and 618 routed through a data cable 608. A number of switches 610 and lights 612 located on the digital audio mixer console 650 are connected to the mixer control board 604. Control actuators (not separately shown), associated with the mixer control board 604, automatically position and/or lock the location of a particular type of switch, such as a linear slider, a push button, or a rotary knob (not separately shown). A Musical Instrument Device Interface ("MIDI") port 614 is coupled to the computer system.

The digital audio mixer control board 604 is coupled to a liquid crystal display ("LCD") 616 that provides a user with a visual indication of the status and/or operation of controls on the digital audio mixer console 650. In other embodiments, a vacuum fluorescent display or light emitting diode ("LED") display may be used in place of LCD 616. Switches 610 and lights 612 further provide the user with visual indications of the status and/or operational information related to the state of the digital audio mixer control board.

The serial bus 618 that couples the computer system 602 with the master DSP 606 is a special high speed serial bus that is capable of operating at 900 kilobaud (currently running at 115 kilobaud in a preferred embodiment). Alternatively, a universal serial bus ("USB") or FireWire communication link could be employed to couple the computer system 602 to the master DSP 606. The master DSP 606 is located on a DSP board 620 that also includes a number of slave DSPs 622 and 624, designated  $S_1$ - $S_N$ , that are coupled in parallel with the master DSP 606 via a DMA

bus 626. FIGURE 1 shows slave DSPs  $S_1$  622 and  $S_N$  624, with a dashed line 628 indicating a number of additional slave DSPs. A preferred embodiment includes a single DSP board 620 that may contain up to a maximum of 48 slave DSPs. Coefficient values for use in determining the signal processing that will be applied by the digital audio mixer to produce the output signals are determined for all of the slave DSPs 622, 624 and 628 by the master DSP 606. These coefficient values are placed by the master DSP 606 into a first set of latches 630 and 632 which are coupled in parallel to DMA bus 626. Each latch is associated with a particular slave DSP. For example, slave DSP  $S_1$  622 is associated with  $S_1$  latch 1 630 and slave DSP  $S_N$  624 is associated with  $S_N$  latch 1 632.

Typically, a slave DSP, such as slave DSP  $S_1$  622, controls an A/D converter ("ADC"), such as ADC 634, that receives an analog signal on an analog input line, such as analog input line 642, and converts the analog signal into a corresponding digital signal that is then input to the slave DSP via a digital signal line, such as digital signal line 644. Once the digital signal has been processed by the slave DSP, a DAC, such as DAC 636, is used to convert the values in the digital signal output from the slave DSP via a digital signal line, such as digital signal line 646, into an analog signal that is supplied on an analog output line, such as analog output line 648. Slave DSP  $S_N$  624 receives digital input 645 from an associated ADC 638 and outputs an output signal 647 to a DAC 640. The digital output signals of slave DSPs are also stored in a second set of latches associated with the DSPs. For example, the digital output signal of slave DSP  $S_1$  is stored in  $S_1$  latch 2 631. Alternative embodiments provide for a number of DSP boards coupled to a computer system in order to increase the number of slave DSPs for processing analog signals.

A channel comprises an input signal and an output signal, such as analog input signal 642 and analog output signal 648 that are controlled by a single slave DSP. A channel's input signal may be provided in a digital format that does not require A/D conversion prior to processing by the slave DSP. In such cases, an ADC is not required for converting the input to a digital signal. For example,



ADC 634 would not be provided to convert the input signal 642 from analog to digital if input signal 642 were in a digital format. A channel may produce a digital signal output. For example, if a digital output from slave DSP S<sub>1</sub> 622 were desired, then DAC 636 would not be included to convert the output signal from digital to analog. However, additional circuitry may be provided to enable slave DSPs to input and output a particular type of the digital signal, e.g., circuitry for processing Alesis Corporation's Alesis Digital Audio Tape ("ADAT"), Tascam Corporation's Tascam Digital Interface Format ("TDIF"), and the Audio Engineers Society/European Broadcast Union ("AES/EBU") formats. Although the embodiment shown in FIGURE 1 provides signal converters for every input and output, other embodiments may be differently configured. Selected channels may provide A/D input-only conversion, D/A output-only conversion, or no input or output conversion.

The computer system 602 may optionally be coupled to a remote computer system (not shown) so that additional services can be remotely provided or so that the audio mixer 600 may be remotely controlled. Additional services may include updating software executed by the computer system 602, and may also include enabling a technical service representative to troubleshoot a problem with the digital audio mixer console 650 from a geographically remote or distant location.

A video display 652, a keyboard 654, and a pointing device 656 are also preferably coupled to the computer system 602, so that the user may control the operation of the digital audio via a visual console displayed on the video display 652. The control actuators of the digital audio mixer control board 604 are responsible for updating the lights 612 and switches 610 of the digital audio mixer console 650 to reflect any changes input by the user via the virtual console. The user may employ the pointing device 656 and/or the keyboard 654 to manipulate the virtual console to control input and output signals processed by the digital audio mixer. Alternatively, the user may manually operate the switches 610 of the digital audio mixer console 650 to control the input and output signals of the digital audio mixer.

An accessory card 658 may also be included within the digital audio mixer console 650 to provide additional features, such as selected audio effects, digital outputs, and synchronization signals. The accessory card 658 is coupled to the digital audio mixer control board 604 via a bus 660. A number of accessory boards may be  
5 connected to this bus 660. It is envisioned that third party developers may be licensed to provide accessory cards for the present invention. Although not shown, internal accessory boards may be coupled to the computer system 602 to provide additional features, such as audio effects.

FIGURE 7 is an overview flowchart of the logical steps performed by the  
10 DSP board in response to receiving instructions from the computer system. In step 702, the master DSP 606 receives instructions from the computer system 602. Then, in step 704, the master DSP 606 determines the coefficients specified in the received instructions for each slave DSP. Next, in step 706, the master DSP 606  
15 strobes a number of latches (e.g., latches 630 and 632 in FIGURE 6) that store the new coefficients. These latches are directly coupled to the DMA bus 626 and each is associated with a particular slave DSP. Next, in step 708, the master DSP 606 strobes, or commands, each specified slave DSP to load the new coefficients from the latch that is associated with that slave DSP. After these new coefficients are loaded,  
20 each slave DSP processes input signals according to the new coefficient and provides a new output signal to another latch coupled to the DMA bus 626 that is associated with the slave DSP (e.g., latches 631 and 633 in FIGURE 6). Finally, in step 710, each slave DSP is strobed or commanded to store certain of the new output signals from the latches produced by specified slave DSPs into memory locations within the  
25 slave DSP that correspond to the specified slave DSPs that produced the output signal.

FIGURE 8 shows an exemplary computer system incorporated in one embodiment of a digital audio mixer. The computer system 800 includes CPU motherboard 802, a processor 804, a memory 806, an Industry Standard Architecture ("ISA") bus 808, and a Peripheral Component Interconnect ("PCI") bus 810. A PCI

bus controller 812 and an ISA bus controller 814 are employed by processor 804 to monitor and control PCI bus 810 and ISA bus 808, respectively. Most newer PCI bus controllers implement the ISA bus behind the PCI bus, so that the processor 804 accesses the ISA bus 808 via the PCI bus controller 812.

5       Memory 806 includes read only memory ("ROM") 816 and random access memory ("RAM") 818. The basic routines that are employed to transfer information between the components of the CPU motherboard 802 are included in a basic input/output system ("BIOS") 820, which is stored in ROM 816. An operating system 822 and a software program 824 are loaded into RAM 818 before the  
10       software program 824 is executed by the processor 804. The operating system 822 may include a third party kernel. In a preferred embodiment, the operating system kernel will support real time control and preemptive tasking. Certain aspects of the present invention are implemented by operating system 822 using a number of routines provided by software program 824. Other programs that may also be loaded  
15       into RAM 818 include device drivers that are associated with the devices coupled to CPU motherboard 802 through either the ISA bus 808 or the PCI bus 810.

      CPU motherboard 802 is optionally coupled to a hard disk drive 826, which can be used to store software programs, script files, and data. Although not required, the CPU motherboard 802 may also be coupled to access the memory on other  
20       nonvolatile storage devices such as a magnetic disk drive 828, and an optical disk drive 830. These nonvolatile storage devices are coupled to either PCI bus 810 or ISA bus 808 by a hard disk drive interface 832, a magnetic disk drive interface 834, and an optical disk drive interface 836. Magnetic disk drive 828 is employed for reading and writing to a removable magnetic disk 838, while optical disk drive 830  
25       provides for reading from a removable optical disk 840, such as a Compact Drive-Read Only Memory ("CD-ROM") disk, digital versatile/video disk ("DVD"), or other optical media. The combination of one of drives 828 and 830 with media 838 and 840, respectively, provides for the nonvolatile storage of computer readable instructions and other data for use by processor 804. In a preferred embodiment,

different types of data may be stored by hard disk drive 826, removable magnetic disk 838, optical disk 840, ROM 816, and RAM 818. Furthermore, other types of media, such as magnetic cassettes and flash memory cards (not shown), may be employed by CPU motherboard 802 for storing data.

5           A user of the computer system 800 may employ input devices, such as a pointing device (mouse) 842 and keyboard 844, to input commands and/or data. Although not shown, other types of input devices may also be included in the computer system, including a microphone, joystick, game pad, modem, satellite dish, and scanner. Input devices are coupled by an input/output (I/O) device interface 846  
10   to processor 804 through either PCI bus 810 or ISA bus 808. Different I/O device interfaces (not separately shown) may be employed to provide various ports, such as serial, parallel, game, small computer serial interface ("SCSI"), USB, and FireWire ports, for connecting different types of input/output devices to the computer system 800. For instance, the digital audio mixer control board 604 (FIGURE 6)  
15   may be coupled to a port using an I/O interface that supports the port. Also, interfaces for one or more ADCs and DACs not included on the DSP Board 620 (FIGURE 6), digital input/output encoders, video frame grabbers, and other equipment may optionally be included. Display 848 is connected to the computer system 800 through a video adapter 850 that is coupled to processor 804 through  
20   either PCI bus 810 or ISA bus 808.

          In FIGURE 8, the computer system is shown operating in a networked environment that employs logical connections to one or more remote computers, such as a remote computer 852, which may be a server, a router, a network personal computer ("NETPC"), or another computer system similar to computer system 800,  
25   and which will typically include many of the elements of the computer system 800, described above. A network interface 854 is coupled to processor 804 through either ISA bus 808 or PCI bus 810. Network interface 854 is also coupled through a local area network ("LAN") 856 to remote computer 852. The LAN 856 is employed to provide the connection between the computer system 800 and a networked

environment that may include an office network, an Intranet, and the Internet. In the networked environment, programs that are executable by the computer system 800 may be stored in the memory of the remote computer 852. Moreover, it will be appreciated by those skilled in the art that other means, such as a modem 858, may be employed to establish a connection, such as through a wide area network ("WAN"), between the computer system and the remote computer 852.

A DSP board 847, 620 in FIGURE 6, is coupled to I/O device interface 846 via a high-speed serial communication bus 860. The DSP board 847 enables the real time processing of audio signals by executing over 3 billion instructions a second. Further, the DSP board 847 can process all signals on all channels in 20 microseconds or less. The DMA bus 626 (shown in FIGURE 6) may be implemented as a 16-bit bus or may be implemented as a 32-bit bus. Also, the bus resolution is compressed to a lower resolution using dithering while retaining signal integrity below the noise floor. For example, the 16-bit DMA bus 626 provides 20-bit resolution using UV22® software licensed from Apogee Corporation.

FIGURE 9 shows an overview of the software employed by the present invention. The operating system 822 employs a kernel 902 that includes a task manager 904 for scheduling pre-emptive tasks according to predetermined priorities. A serial communication module 906 controls the serial transfer of data between the computer system 602 (FIGURE 6) and other components, such as mixer control board 604 (FIGURE 6). A loader 908 enables the execution of compiled code, and device drivers 910 control the operation of I/O devices, e.g., keyboard 654, mouse 656, and video display 652 (FIGURE 6). A BIOS 912 is included within the kernel 902 to control the operation of various subsystems associated with input and output to and from the computer system 602. A file system 914 is included to organize files within the operating environment.

An automation engine 916 interprets a configuration script 918 that defines the type, or model, and the components, such as switches and lights, of the digital audio mixer console 650 (FIGURE 6). Automation engine 916 constantly monitors

the communication with the digital audio mixer console 650 and controls the operation of the computer system 602, DSP board 620, and mixer control board 604 (all shown in FIGURE 6). For example, if the user adjusts a fader switch (not separately shown) that is coupled to digital audio mixer console 650, then automation engine 916 immediately implements code that directs DSP board 620 to process the channel signals in a manner that corresponds to the disposition of the fader switch. When an automatic mode is enabled by the user, automation engine 916 directs mixer control board 604 to employ its control actuators to move switches to a particular location, illuminate specific lights, and to provide instructions to DSP board 620 for the processing analog input signals in accord with the disposition of the switches and controls. Also, when digital audio mixer console 650 is in a manual mode, automation engine 916 provides instructions to DSP board 620 that map the processing of the analog signals to the disposition or state of the switches.

A mixer script engine 920 interprets a desktop script 922 and produces a desktop manager 924 for display to the user. An image of a virtual mixer console 926 is produced in the initial window displayed to the user in the desktop manager 924. An event manager 928 handles the display in the desktop manager environment, e.g., the position and selection of objects in the image of virtual mixer console 926. Automation engine 916 implements the highest priority tasks and event manager 928 asserts the lowest priority tasks. In this way, the audio signal processing tasks implemented by digital audio mixer console 650 are always given a higher priority than displaying the image of virtual mixer console 926. The highest priority task is associated with user input, which may occur if the user selects objects in the image of virtual mixer console 926 or moves any of the switches. In any case, no more than a 10-millisecond latency occurs before an analog signal is processed by the components of digital audio mixer console 650.

FIGURE 10 shows an overview of an automatic training sequence ("ATS") executed by the automation engine implemented in accordance with one embodiment of the present invention. In step 1002, the ATS determines if there is a switch map

and an LED map available for use by automation engine 916. These maps provide addresses that correspond to the actual switches and lights coupled to the digital audio mixer console 650 (FIGURE 6). If true, the automatic training sequence logic is terminated. Otherwise, in step 1004, the ATS polls the mixer control board 604 for each address that corresponds to a switch in the digital audio mixer console 650. In step 1006, the ATS builds a switch map that associates each such switch with a corresponding address. In step 1008, the ATS polls the mixer control board 604 for each address that corresponds to an indicator light of the mixer console 650 coupled to the mixer control board 604. In step 1010, the ATS builds an LED map that associates each indicator light or LED with a corresponding address, after which the ATS returns.

It is envisioned that the initial training sequence for the automatic engine 916 could also be performed manually by the user. For example, the user of digital audio mixer console 650 could manually actuate every switch and control disposed on the console, so that automation engine 916 could sense the presence (address) for each device, including the indicator lights.

FIGURE 11 shows an overview of the startup sequence executed by the automation engine to initialize the digital audio mixer. In step 1102, the startup sequence loads a switch map that identifies the corresponding address for each switch disposed in digital audio mixer console 650 (FIGURE 6). In step 1104, the startup sequence loads the LED map which identifies the corresponding address for each indicator light on the mixing console. In step 1106, the startup sequence directs the automation engine 916 to interpret configuration script 918. The configuration script contains indications of types of features and functions and corresponding operations to be performed on the indicated features and functions, thereby enabling and disabling operational features of the mixing console 650. For example, indications to enable a surround sound mixing control and to disable an equalizer control may be contained in the configuration script 918.

In step 1108, the startup sequence directs the automation engine 916 to create an automation engine buffer for storing a number of values, such as preset configurations. In step 1110, the startup sequence directs the mixer script engine 920 to interpret desktop script 922, which is employed to define the image presented to the user in the initial window of desktop manager 924. The desktop script 922 specifies the virtual switches, lights, and controls to be displayed to the user in a particular window. The displayed objects in each window are graphical representations of the corresponding elements of mixing console 650, and show the states of those elements.

Finally, in step 1112, the startup sequence directs the desktop manager 924 to display the image of the initial window to the user. In a preferred embodiment, the initial window includes the image of virtual mixer console 926. However, the initial window may also be configured to provide software controls that enable the user to select a different window that displays the image of the virtual mixer console 926 or additional mixer console features.

FIGURE 12 shows further details of configuration script interpretation by the automation engine. In step 1202, automation engine 916 sets a type for the digital audio mixer console 650, which is identified in the configuration script. In step 1204, the automation engine 916 sets an address for each switch and light. In step 1206, the automation engine 916 sets the name for each addressed switch and indicator light. In step 1208, the automation engine 916 sets a maximum value for each switch. In step 1210, the automation engine 916 sets the initial value, or state, for each switch and indicator light. Typically, the indicator light and switch initial values are zero. Finally, in step 1212, the automation engine 916 designates each switch and indicator light as either an automatic or non-automatic device.

FIGURES 13A, 13B, and 13C show an exemplary desktop script employed by mixer script engine 920. The script text may be edited with a text editor, such as Microsoft Corporation's NOTEPAD™ editor. An important feature of the present invention is the use of a script that employs real language text to define the operation



of digital audio mixer console 650 and to define the images displayed to the user. Also, since the scripts are interpreted, they may be updated on the fly without requiring a reboot of the computer system or recompilation and reloading of software components.

5 It is envisioned that the user of a digital audio mixer may obtain updated configuration and desktop scripts to correct software bugs and to provide new features for digital audio mixer console 650. These updates may be provided through a network connection to remote computer, or over the Internet, or may be stored and transferred from a physical media, such as a floppy disk.

10 Script updates are preferably encrypted to prevent unauthorized changes in the operation and control of the mixer console. In a preferred embodiment, RSA encryption is used and the serial number of digital audio mixer console 650 is employed as the public encryption key. In this manner, script updates can be provided that are limited for use with a particular mixer console.

15 It is also contemplated that an editor may be provided for the user to edit the desktop and/or configuration scripts. The editor would enable the user to customize the look and feel of the images of the virtual mixer console 926 (FIGURE 9) and enable/disable features of the digital audio mixer console 650. Also, it is envisioned that new combinations of features could be developed by the user with the editor.

20 A look-up table is employed by mixer script engine 920 (FIGURE 9) to build the real language text displayed to the user. The software enables the user to select any one of a number of different languages in the look-up table as the real language text. Also, the software will automatically clip out of range parameters in a script and define hot keys. Additionally, the ordering of the text in a script determines  
25 screen overlap for objects and/or windows defined by the text. Further, scripts can be nested in calls from selected objects in the initial window produced for the desktop manager.

In the present invention, the user may employ a "swipe command" to instantaneously change the state of a number of switches with the passage of a cursor

over objects, i.e., virtual switches, displayed in the image of virtual mixer console 926. Preset configurations for controls associated with digital audio mixer console 650 may be stored and enabled, or the configurations may be changed.

A number of different audio effects may also be implemented by the software.

5 For example, the center point of a number of channels for a surround sound mix may be selected with a software control. Two different center points may be selected so that morphing and flyback effects may be produced. A number of addressable software controls may be configured to correspond to a single channel. A number of script templates may be provided to the user for enabling different features and  
10 displays.

Digital audio mixers designed according to alternative embodiments of the present invention may be used for processing audio signals in radio and television studios, live performances, and public address systems. Such digital audio mixers can be employed to control a number of lights in a studio or during a live  
15 performance. The architecture of the present invention specifies systems that are software configurable, so that they can be used for many other applications besides use as a mixing console. A digital audio mixer designed according to the present invention is easily reconfigurable and upgradeable, since the architecture is readily extensible.

20 Although the present invention has been described in connection with the preferred form of practicing it, those of ordinary skill in the art will understand that many modifications can be made thereto within the scope of the claims that follow. Accordingly, it is not intended that the scope of the invention in any way be limited by the above description, but instead be determined entirely by reference to the  
25 claims that follow.

## CLAIMS

1. A digital signal mixer that electronically blends and enhances input signals to produce output signals, the digital signal mixer comprising:

a number of inputs that each receives an input signal and produces an internal digital signal;

a number of digital signal processor components that each processes a number of internal digital signals, received from a number of inputs, to produce a number of processed internal digital signals, each digital signal processor component subsequently processing a number of processed internal digital signals to produce a number of internal output digital signals;

a bus that interconnects the slave digital signal processor components and that allows the slave digital signal processor components to exchange processed internal digital signals; and

a number of outputs that each outputs an output signal corresponding to an internal output digital signal received from a slave digital signal processor component.

2. The digital signal mixer of claim 1 wherein an input that receives an analog signal includes an analog-to-digital converter that converts the received analog signal to an internal digital signal.

3. The digital signal mixer of claim 1 wherein an output that outputs an analog signal includes a digital-to-analog converter that converts the internal output digital signal received from a slave digital signal processor component to an analog output signal.

4. The digital signal mixer of claim 1 wherein a slave digital signal processor component includes:

a digital signal processing logic circuit that processes internal input digital signals to produce processed internal input digital signals;

a memory that stores processed internal input digital signals produced by the digital signal processing logic circuit as well as by digital signal processing logic circuits in different digital signal processor components; and

a mixing logic circuit that blends processed internal input digital signals stored in the memory to produce internal output digital signals.

5. The digital signal mixer of claim 4 further including:

a master digital signal processor that is interconnected with the digital signal processor components via the bus; and

a computer system coupled to the master digital signal processor via a communications link, the remote computer system providing a virtual control panel that includes virtual control elements that control the mixing and processing of signals by the digital signal processor, the computer downloading control information to the master digital signal processor via the communications link, the master digital signal processor downloading control information to the digital signal processor components via the bus, the master digital signal processor controlling operation of, and exchange of information between, the digital signal processor components.

6. The digital signal mixer of claim 5 wherein each input is paired with an output, an input/output pair composing a channel.

7. The digital signal mixer of claim 6 wherein each digital signal processor component can process signals corresponding to up to 8 channels and the digital signal mixer may include up to 24 digital signal processor components.

8. The digital signal mixer of claim 7 combined with a higher-level master digital signal processor and a second digital signal mixer to provide a digital signal mixing system that may include up to 48 digital signal processor components.

9. The digital signal mixer of claim 5 wherein the computer system runs scripts that specify the appearance and operation of the virtual control panel, the scripts generated on the computer system by a user via an editing program or generated by a user on a remote computer using an editor program and downloaded via a network to the computer system.

10. The digital signal mixer of claim 1 wherein the digital signal mixer receives input audio signals and outputs output audio signals.

11. The digital signal mixer of claim 10 wherein the input audio signals include audio signals produced by audio sources selected from among:

- microphones;
- musical instruments;
- synthesizers;
- audio signal receivers;
- computer systems that produce audio signals; and
- recorded music playback devices.

12. The digital signal mixer of claim 10 wherein the output audio signals are output to audio signal processing components selected from among:

- audio speakers;
- audio recording devices;
- audio signal broadcast devices; and
- audio computer systems that can receive and process audio signals.

13. A method for mixing and enhancing digital signals within a digital signal mixer by successively:

receiving input signals;

producing internal digital signals corresponding to the received input signals, a separate internal digital signal produced from each received input signal;

processing the internal digital signals by digital signal processor components to produce processed internal digital signals, a separate processed internal digital signal produced from each internal digital signal, and storing each processed internal digital signal into a memory device;

exchanging stored processed internal digital signals between digital signal processor components so that each digital signal processing component may access processed internal digital signals produced by any other digital signal processing component;

mixing the stored processed internal digital signals to produce internal output digital signals, each internal output digital signal generated from a single stored processed internal digital signal or from a combination of a number of stored processed internal digital signals; and

outputting the internal output digital signals.

14. The method of claim 13 wherein receiving the input signals, producing the internal digital signals, processing the internal digital signals, mixing the stored processed internal digital signals, and outputting the internal output signals is successively repeated at a sufficient rate that, when analog audio signals are output, they are perceived to be continuous audio signals by a human listener.

15. The method of claim 14 wherein, when analog signals are received as input signals, the received analog signals are converted to internal digital signals by an analog-to-digital converter.

16. The method of claim 15 wherein, when analog output signals are desired, the internal output signals are converted to analog output signals by a digital-to-analog converter.
17. The method of claim 14 wherein processing an internal digital signal by a digital signal processor component to produce a processed internal digital signal includes equalizing the signal.
18. The method of claim 14 wherein processing an internal digital signal by a digital signal processor component to produce a processed internal digital signal includes compressing the signal.
19. The method of claim 14 wherein the received input signals are audio signals generated by audio signal generating devices selected from among microphones, musical instruments, synthesizers, audio signal receivers, computer systems that produce audio signals, and recorded music playback devices.
20. The method of claim 14 wherein the output signals are audio output signals and are transmitted to audio signal consuming devices selected from among audio speakers, audio recording devices, audio signal broadcast devices, and audio computer systems that can receive and process audio signals.
21. The method of claim 14 further including:
- controlling the processing of internal digital signals, the exchange of processed internal digital signals, and the mixing of processed internal digital signals by the digital signal processor components by issuing instructions from a master digital signal processor; and
  - displaying on a computer system connected to the master digital signal processor a virtual console that allows a user to control the processing and mixing

operations by interpreting user input, generating instructions corresponding to the user input, and transmitting the instructions from the computer system to the master digital signal processor.

22. The method of claim 21 wherein the display of the virtual console is controlled by running one or more scripts within the computer system.

23. The method of claim 21 wherein the scripts are created and edited by a user, in the form of plain text scripts, on the computer system.

24. The method of claim 21 wherein the scripts are created and edited by a user, in the form of plain text scripts, on a remote computer system and transmitted to the computer system via a computer network.

25. A system for processing and mixing audio signals, comprising:

(a) a plurality of audio channels, each audio channel being coupled to an audio signal input, including:

(i) a converter having an input coupled to receive an audio signal over the audio channel and having an output on which said converter produces a digital signal corresponding to the audio signal; and

(ii) a signal processor coupled to the converter, said signal processor processing the digital signal as a function of at least one predefined coefficient and producing a processed digital signal;

(b) a controller device coupled to the signal processor to enable a user to control the plurality of audio channels;

(c) a computer; and

(d) a script that specifies a plurality of logical steps for execution by the computer to define an operation of the signal processor and the controller, the computer executing the logical steps to process and mix audio signals input on



the plurality of audio channels in real time.

26. The system of claim 25, wherein the converter is a digital to digital converter that receives the audio signal in a specific digital format and produces the digital signal in a different digital format that is compatible for input to the signal processor.

27. The system of claim 25, wherein the converter is an analog to digital converter that receives the audio signal in an analog format and produces the digital signal.

28. The system of claim 25, wherein the audio channel further comprises another converter coupled to the signal processor to receive the processed digital signal, the other converter producing an audio signal output.

29. The system of claim 28, wherein the other converter is a digital to analog converter that produces the audio signal output in an analog format.

30. The system of claim 28, wherein the other converter is a digital to digital converter that produces the audio signal output in a digital format different than that of the processed digital signal.

31. The system of claim 25, further comprising an editor that enables the user to edit the script.

32. The system of claim 25, further comprising a ditherer coupled to the signal processor to dither the processed digital signal to increase its resolution.

33. The system of claim 25, wherein the signal processor periodically produces

the processed digital signal, with a periodicity of no more than 20 microseconds.

34. The system of claim 25, further comprising a card bus adapted to receive and couple to one or more digital cards, enabling at least one such digital card to be selectively added to the system.

35. The system of claim 34, further comprising a plurality of digital cards adapted to couple with the card bus, each of the plurality of digital cards providing at least one functional feature, including at least one of: an effect, a MIDI output, a sync signal, a digital output, an alternate input, and an alternate output.

36. The system of claim 25, wherein the signal processor comprises a slave processor that processes the digital signal as a function of at least one predefined coefficient to produce the processed digital signal on a data bus, said system further comprising a master processor that controls the slave processor by determining the predefined coefficient, the master processor being coupled to the slave processor via the data bus.

37. The system of claim 36, wherein the data bus comprises a direct memory access bus.

38. The system of claim 36, further comprising a latch that is associated with the slave processor and which enables the processed digital signal produced by the slave processor to be output on the data bus.

39. The system of claim 36, wherein each audio channel has a plurality of slave processors controlled by the master processor, each slave processor having a memory for storing each processed digital signal produced by each of the slave processors, said master processor determining predefined coefficients for each of

the plurality of slave processors.

40. The system of claim 36, wherein the signal processor executes at least three billion instructions per second.

41. The system of claim 25, wherein the computer is linked in data communication with a network.

42. The system of claim 41, further comprising a modem adapted for coupling the computer in data communication to the network through a telephone line.

43. The system of claim 41, further comprising a network interface adapted to couple the computer in data communication with the network.

44. The system of claim 43, wherein the network interface further comprises a hub.

45. The system of claim 41, wherein the data communication is encrypted.

46. The system of claim 41, wherein the data communication enables a different script to be downloaded from a site over the network to the computer, the different script including other logical steps that are executed by the computer to control at least one of the plurality of audio channels.

47. The system of claim 46, further comprising a web browser executed on the computer and adapted to enable the user to select the other script from a plurality of scripts available at the site over the network.

48. The system of claim 25, further comprising an operating system executed by the computer, the operating system assigning a time stamp to each of a plurality of events and employing the time stamp of each event to order execution of a preemptive task associated with each event.

49. The system of claim 25, wherein the script comprises a desktop script and a configuration script.

50. The system of claim 25, wherein the script comprises a text file.

51. The system of claim 25, further comprising a display coupled to the computer, said display displaying an image of the controller corresponding to a state of the controller.

52. The system of claim 49, further comprising mixer engine software executed by the computer, the mixer engine software interpreting the desktop script to produce the image on the display.

53. The system of claim 52, wherein an order of the logical steps in the desktop script is employed by the mixer engine software to control the image on the display.

54. The system of claim 52, wherein the desktop script includes a natural language, the mixer engine software including the natural language in the image on the display.

55. The system of claim 54, wherein the natural language comprises one of English, French, German, Italian, Spanish, Chinese, and Japanese.

56. The system of claim 51, further comprising automation engine software, the automation engine software interpreting the configuration script to define an operation of the controller and enabling a state of the controller to be automatically varied.

57. The system of claim 56, wherein the automation engine software responds to the configuration script by defining a range of values for the controller, the automation engine software automatically employing the range to bound a value determined by the state of the controller.

58. The system of claim 56, wherein the automation engine software senses the state of the controller and associates an address defined by the configuration script with the controller.

59. The system of claim 25, wherein the controller comprises at least one of a switch, a mixer, a fader, and a variable potentiometer.

60. The system of claim 25, further comprising an indicator associated with the controller, the indicator includes a light source and a meter.

61. The system of claim 25, further comprising a hot key, the user being enabled to associate the control of at least one of a plurality of controllers with the hot key so that operation of at least one of the plurality of controllers is automatically controlled by the hot key.

62. The system of claim 25, further comprising a plurality of controllers, at least a portion of the plurality of controllers being selectively ganged together for simultaneous operation.

63. The system of claim 25, further comprising a surround sound effect for at least a portion of the plurality of audio channels, the surround sound effect employing a selectable center point in the processed digital signal for each of the plurality of audio channels comprising said portion.

64. The system of claim 63, wherein the controller enables a selection and storage of at least two center points for the surround sound effect.

65. The system of claim 64, wherein the surround sound effect includes an automated transition between at least two different center points.

66. A system for processing and mixing audio signals, comprising:

(a) a plurality of audio channels, each audio channel being coupled to an audio input signal and including:

(i) a converter that converts the audio input signal into a corresponding digital signal;

(ii) a slave processor coupled to the converter, the slave processor processing the digital signal as a function of at least one predefined coefficient, producing a processed digital signal on a data bus, and producing a mixed digital signal; and

(iii) another converter that is coupled to the mixed digital signal, said other converter converting the mixed digital signal to an output signal that corresponds to the processed digital signal;

(b) a master processor coupled to the slave processor via the data bus, said master processor determining at least one predefined coefficient for the slave processor;

(c) a controller adapted to enable a user to control the plurality of audio channels;

(d) a computer; and

(e) a script including a plurality of logical steps executable by the computer to define an operation of the signal processor and the controller to mix and process audio input signals in real time.

67. A method for processing and mixing audio signals, comprising the steps of:

(a) on a plurality of audio channels, converting audio signals into corresponding digital signals;

(b) processing the digital signals with signal processors, as a function of at least one predefined coefficient, to produce corresponding processed digital signals;

(c) enabling a user to control the processing of the digital signals with a controller; and

(d) defining an operation of the signal processors and the controller with a script comprising a plurality of logical steps, to mix and process the audio signals in real time.

68. The method of claim 67, further comprises the step of converting the processed digital signals to analog output signals.

69. The method of claim 67, further comprises the step of converting the processed digital signal to digital output signals in a different format.

70. A real time system for digitally processing and mixing audio signals, comprising:

(a) a computer that includes a memory and a processor;

(b) an input audio signal;

(c) a digital signal processor that processes the input audio signal as a function of at least one predefined coefficient, producing a processed signal;

(d) a mixer control that produces a value employed by the computer to control the digital signal processor;

(e) a script that is implemented by the computer, the script defining the operation of the mixer control and the digital signal processor; and

(f) an output audio signal.

71. The real time system of claim 70 further including:

when the input audio signal is an analog input audio signal, an analog to digital converter that converts the analog input audio signal to a digital input audio signal;

and when the output audio signal is an analog output audio signal, a digital to analog converter that converts the digital output audio signal to the analog output signal.

72. A real time system for digitally processing and mixing audio signals, comprising:

(a) a computer that includes a memory and a processor;

(b) an input audio signal;

(c) a slave digital signal processor that processes the digital signal as a function of at least one predefined coefficient, producing a processed signal;

(d) a master digital signal processor that calculates the predefined coefficient for the slave processor, the master digital signal processor and the slave digital signal processor being coupled to a data bus;

(e) a mixer control that produces a value employed by the computer to control the master digital signal processor;

(f) a script that is implemented by the computer, the script defining the operation of the mixer control, the master digital signal processor, and the slave digital signal processor; and

(g) an output audio signal.



73. The real time system of claim 70 further including:

when the input audio signal is an analog input audio signal, an analog to digital converter that converts the analog input audio signal to a digital input audio signal;

and when the output audio signal is an analog output audio signal, a digital to analog converter that converts the digital output audio signal to the analog output signal.

74. A real time system for digitally processing and mixing audio signals, comprising:

- (a) a computer that includes a memory and a processor;
- (b) a slave digital signal processor that processes a digital signal as a function of at least one predefined coefficient, producing a processed signal;
- (c) a master digital signal processor that calculates the predefined coefficient for the slave processor, the master digital signal processor and the slave digital signal processor being coupled to a data bus;
- (d) a mixer control that produces a value employed by the computer to control the master digital signal processor; and
- (e) a script that is implemented by the computer, the script defining the operation of the mixer control, the master digital signal processor, and the slave digital signal processor, so that the processed signal is provided as an output.

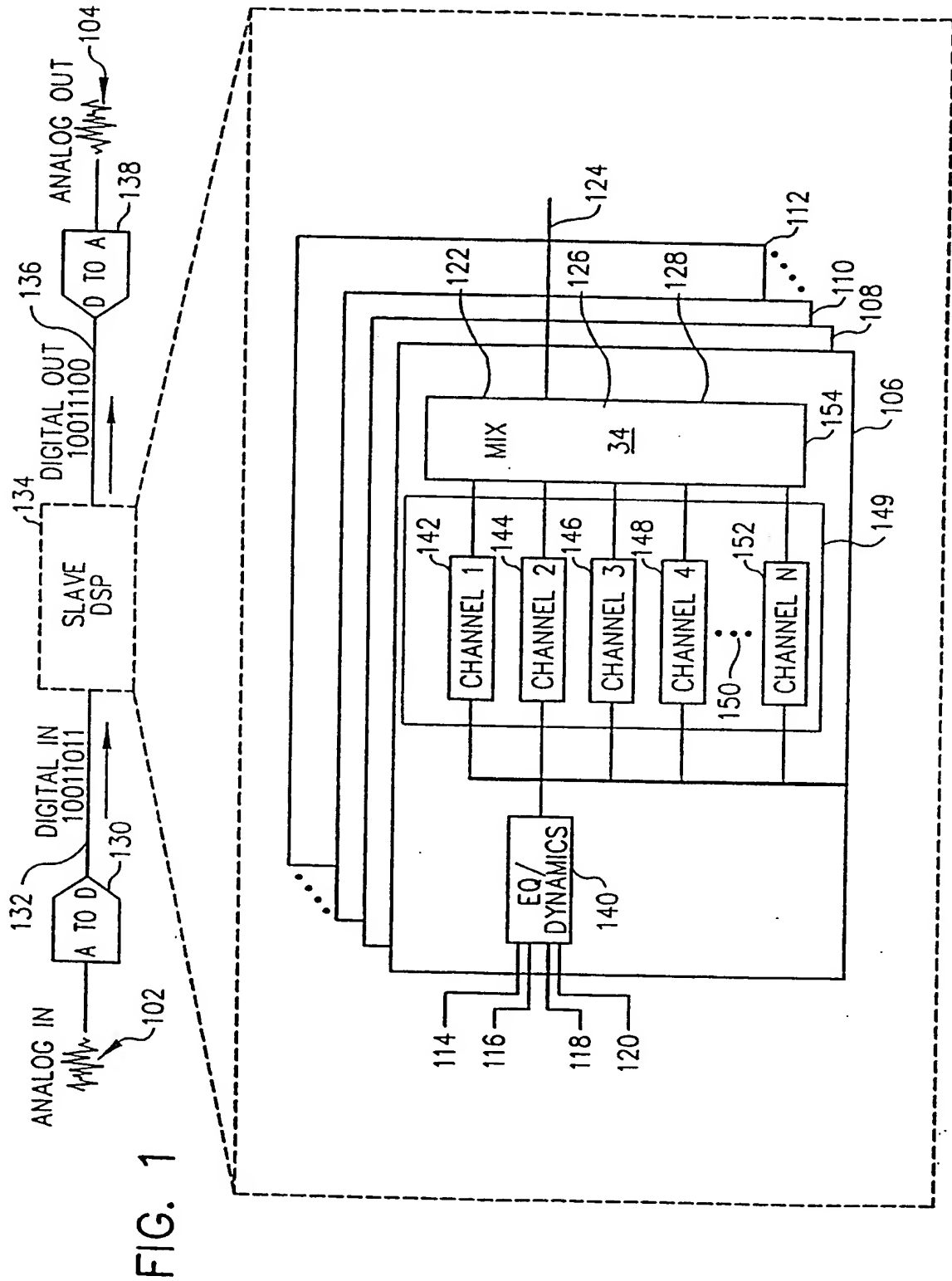


FIG. 2

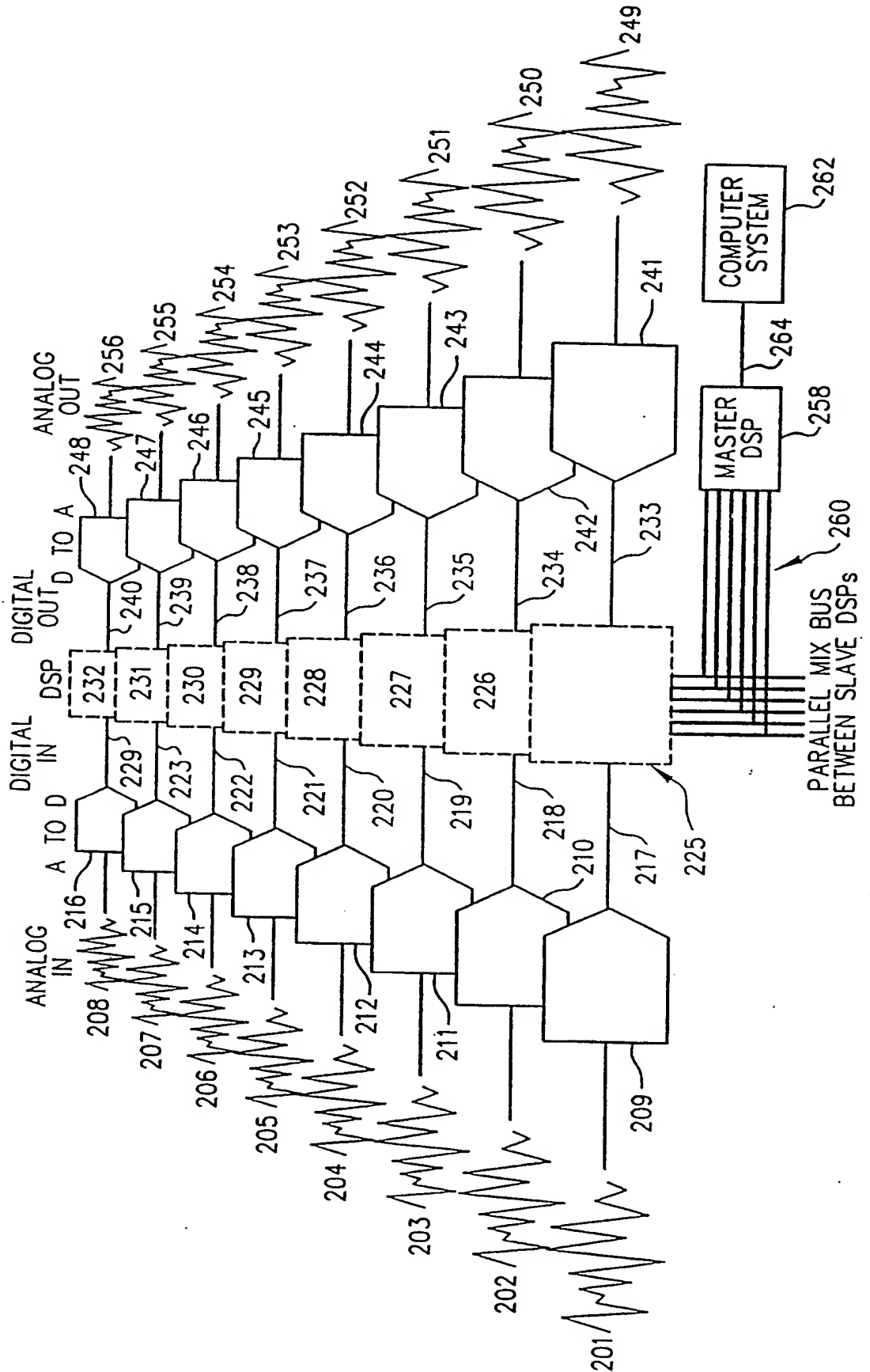


FIG. 3

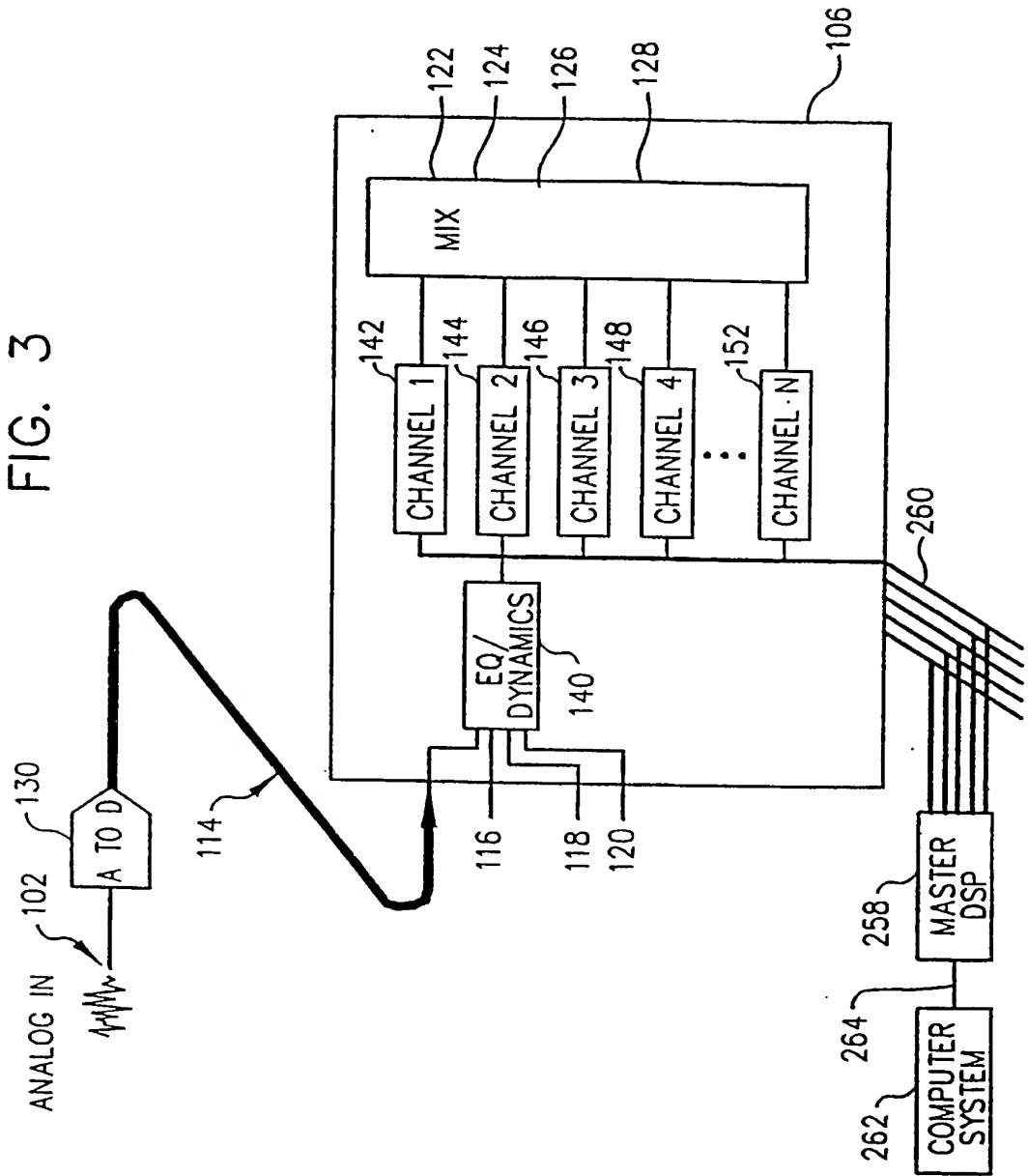
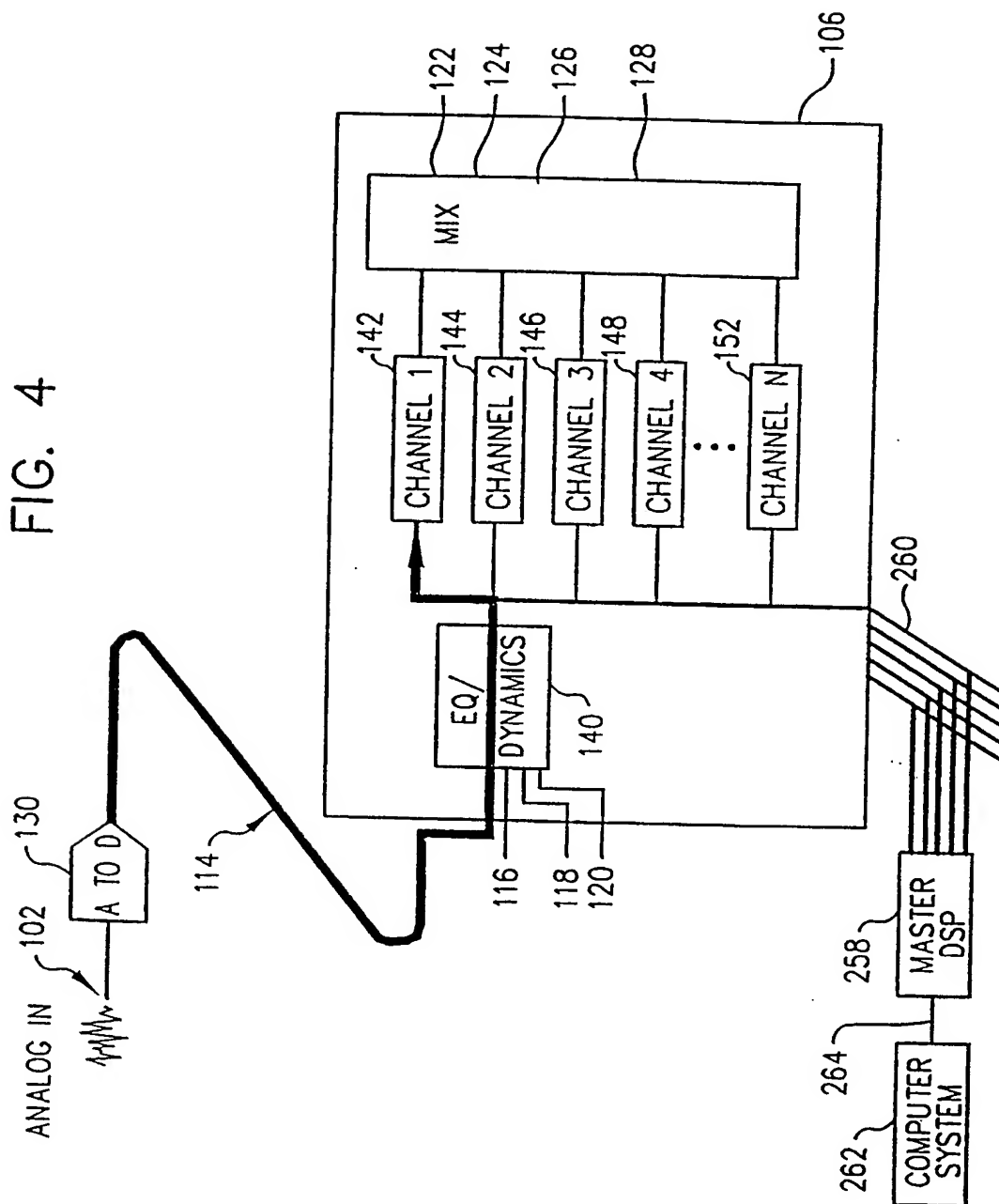
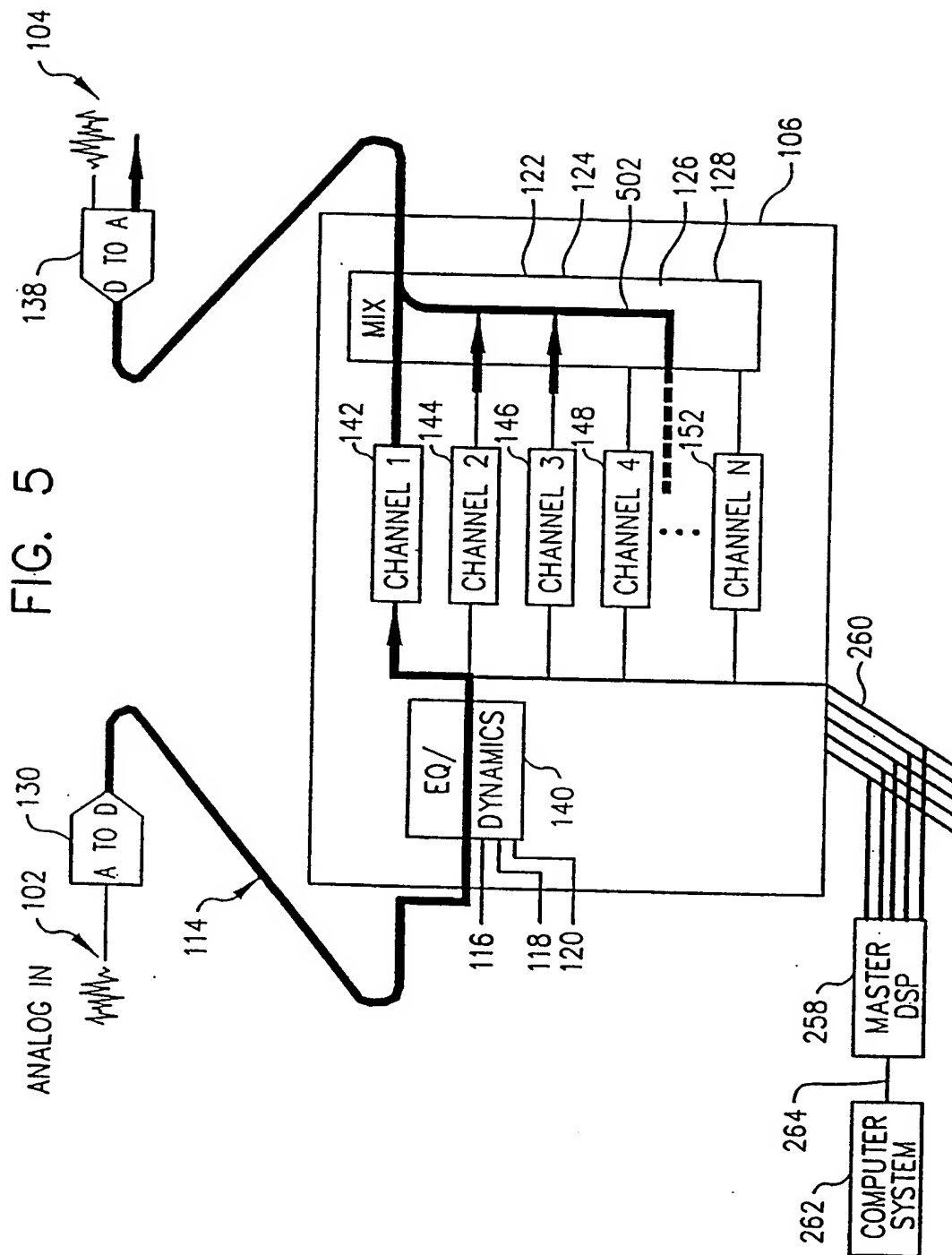
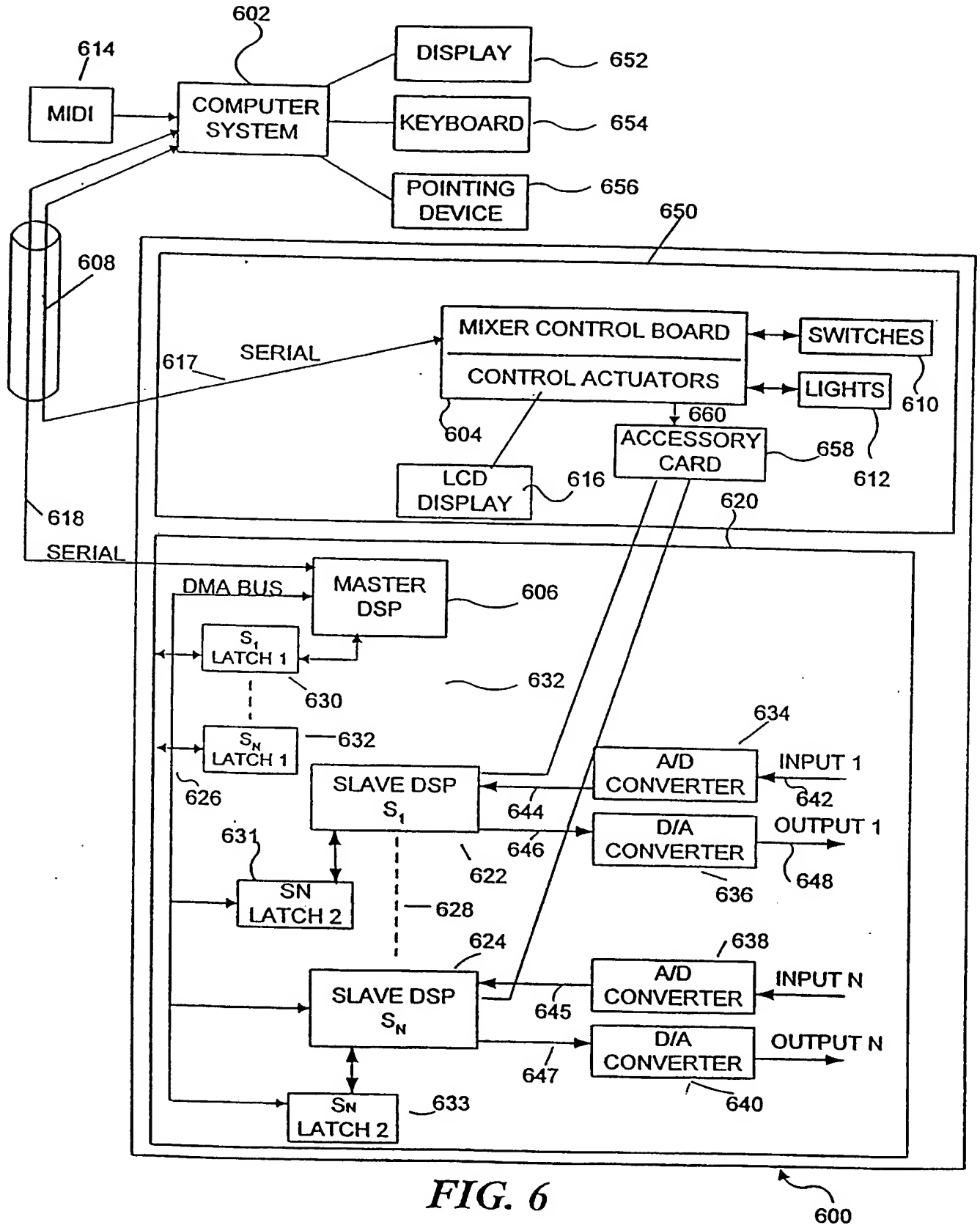


FIG. 4

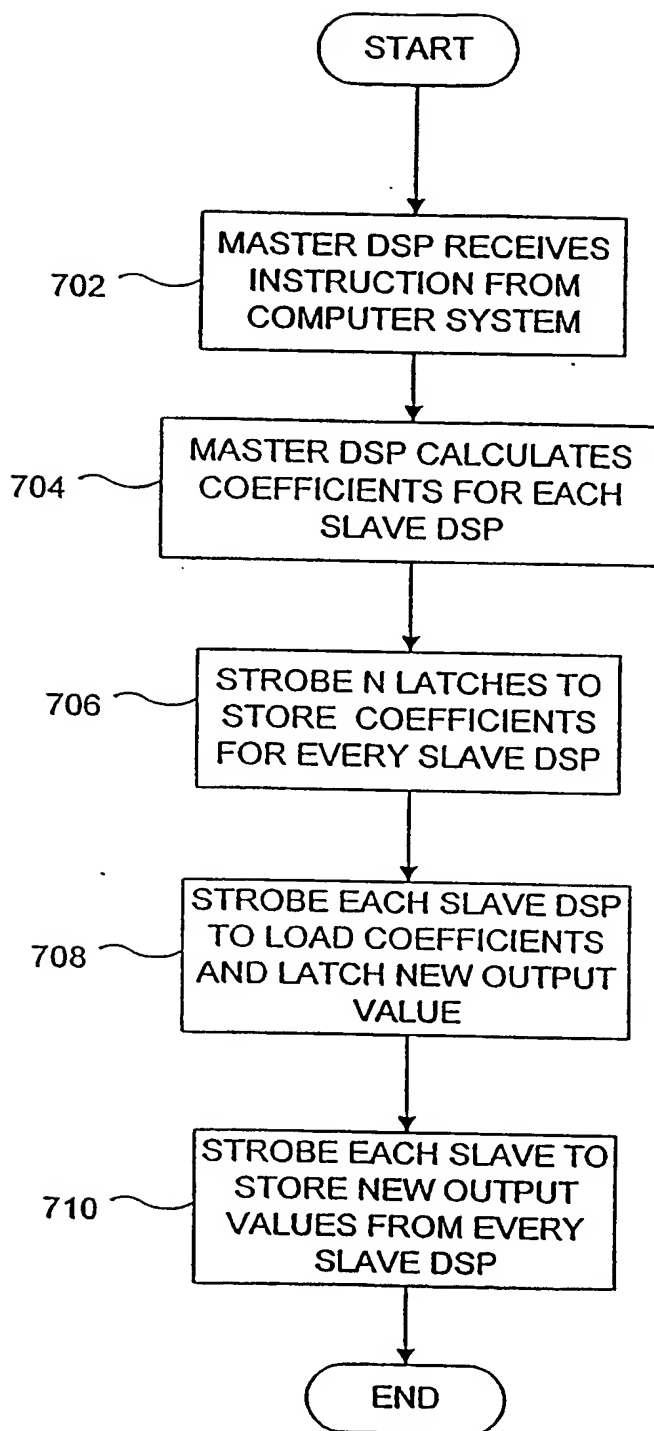




6/14



7/14

**FIG. 7**



8/14

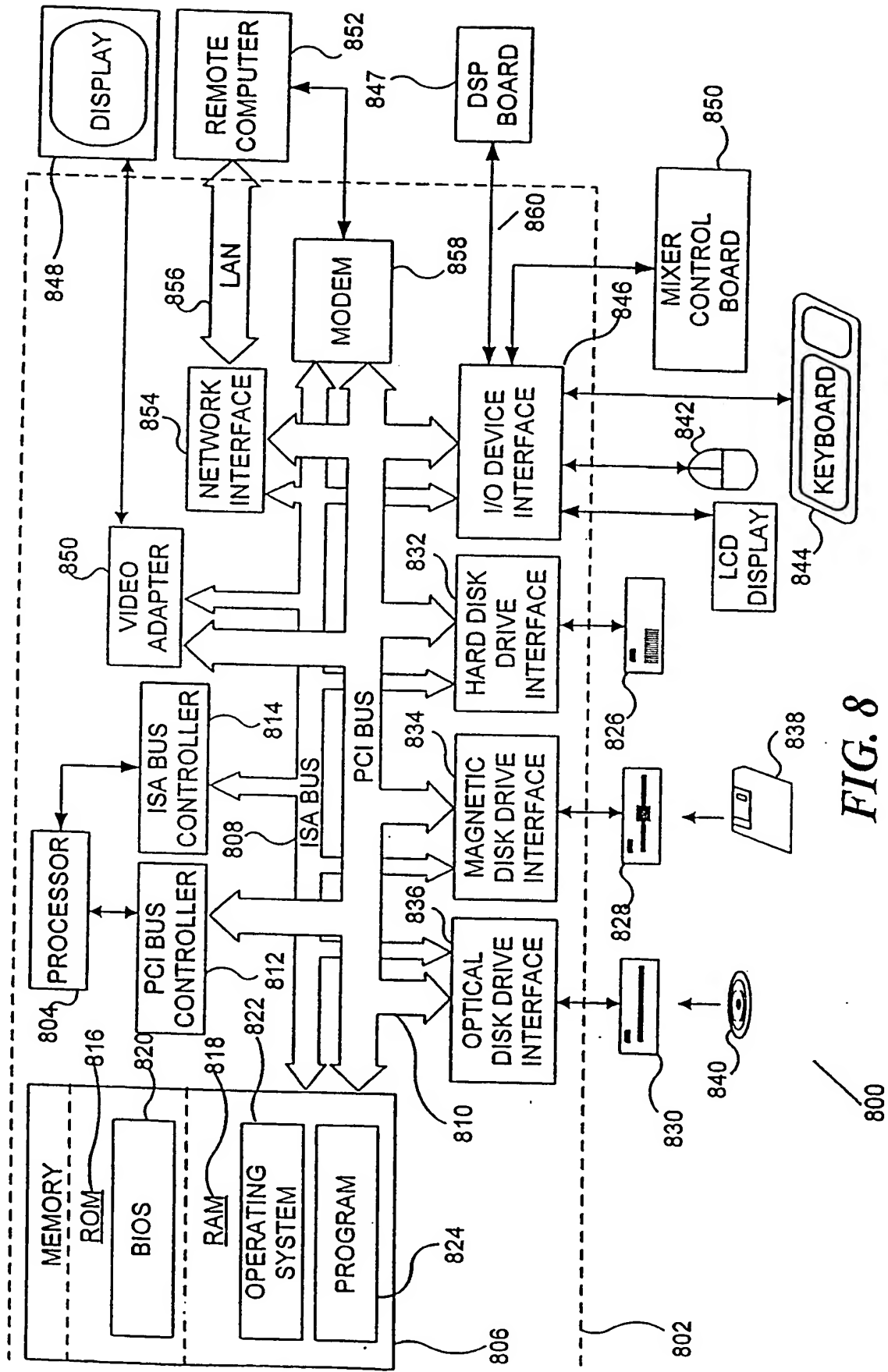
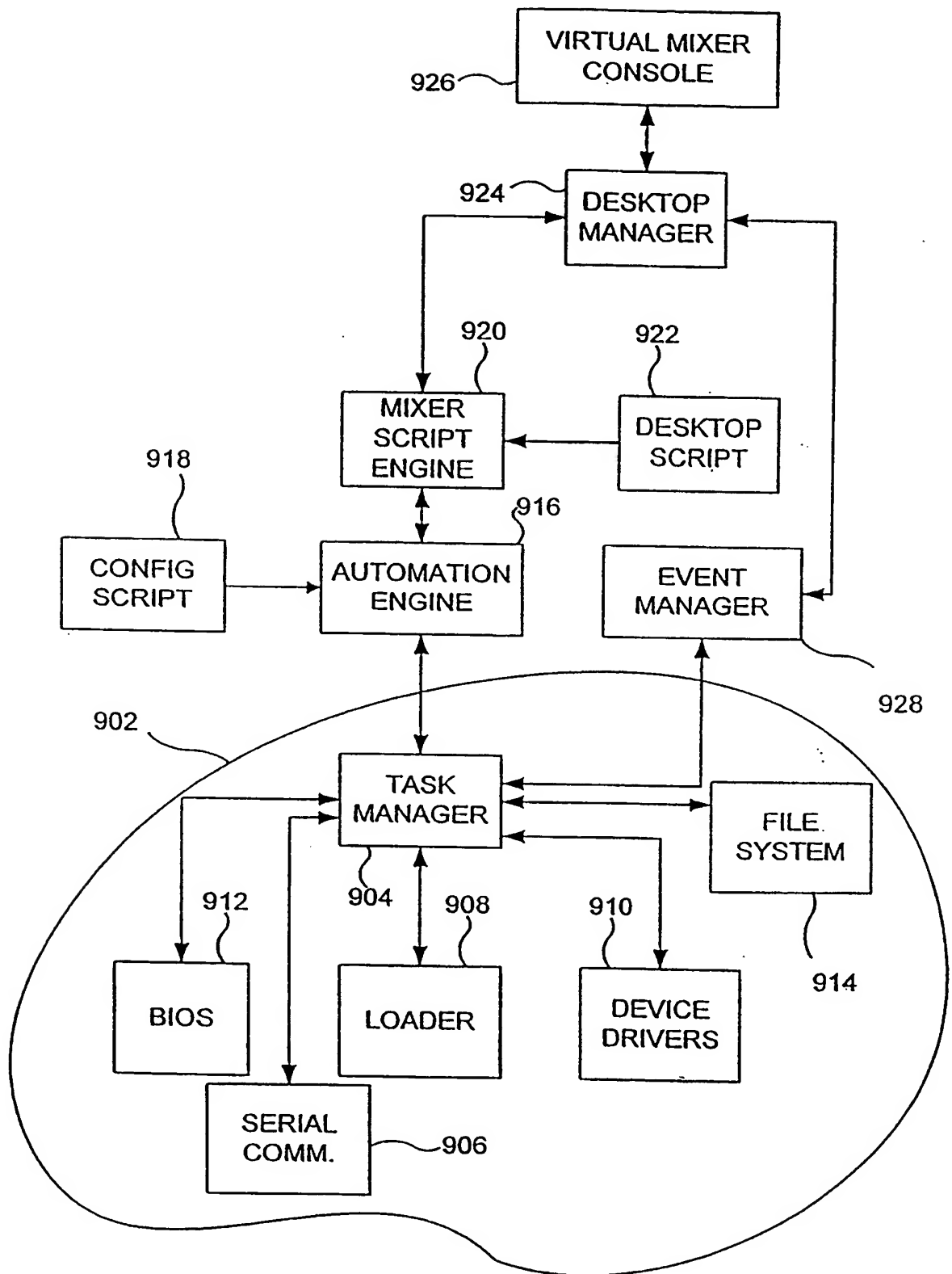
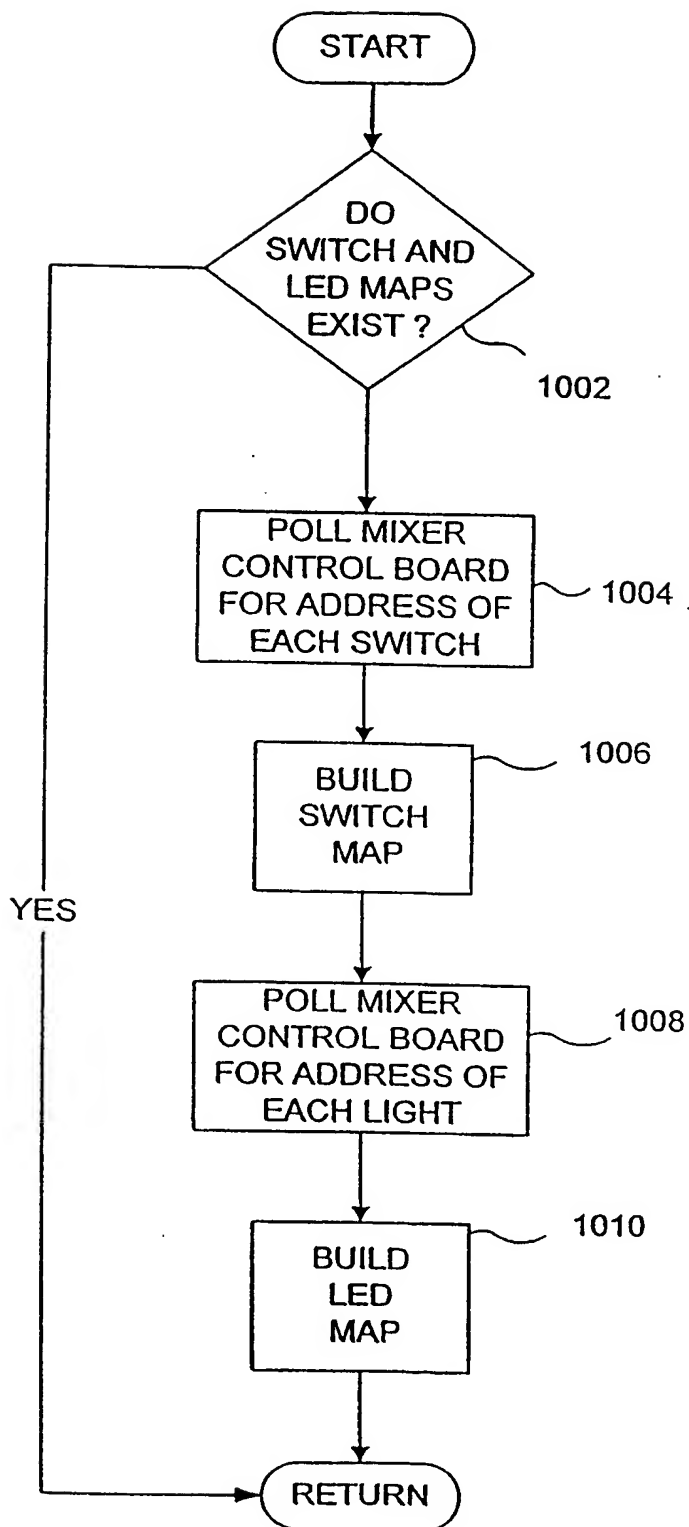


FIG. 8

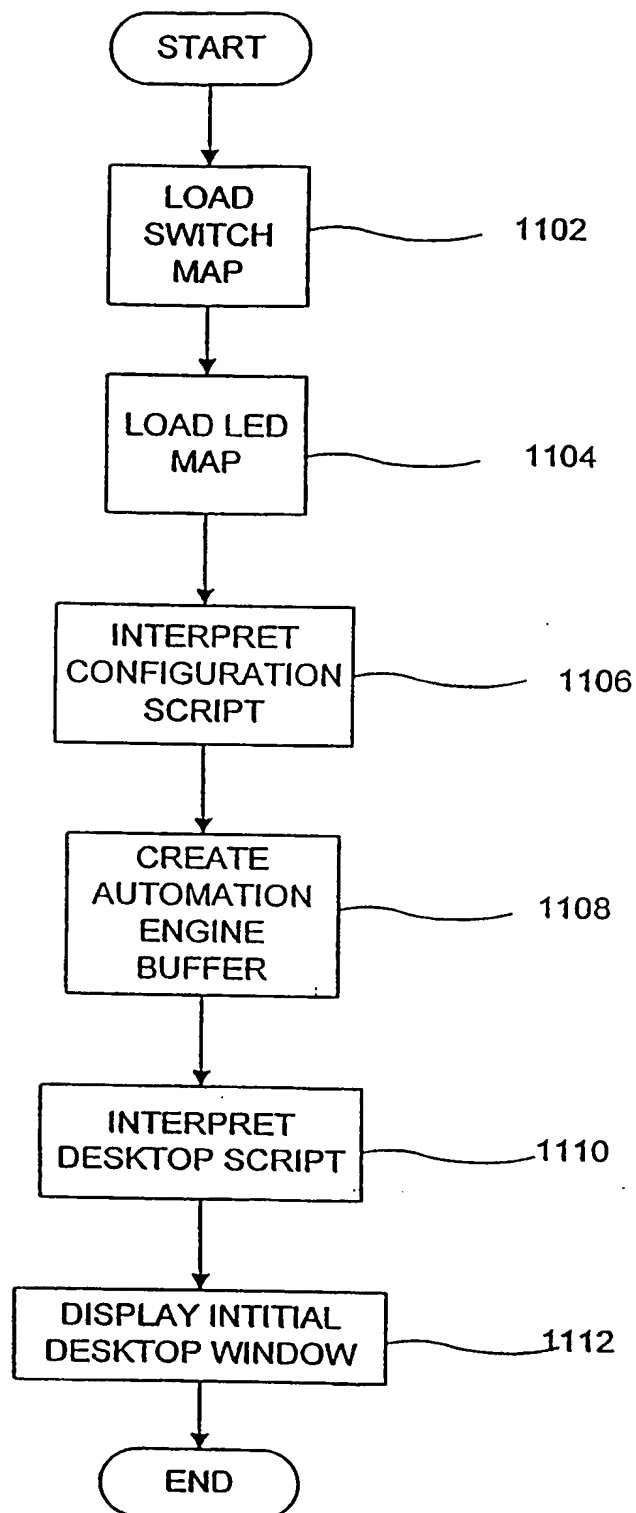
9/14

**FIG. 9**

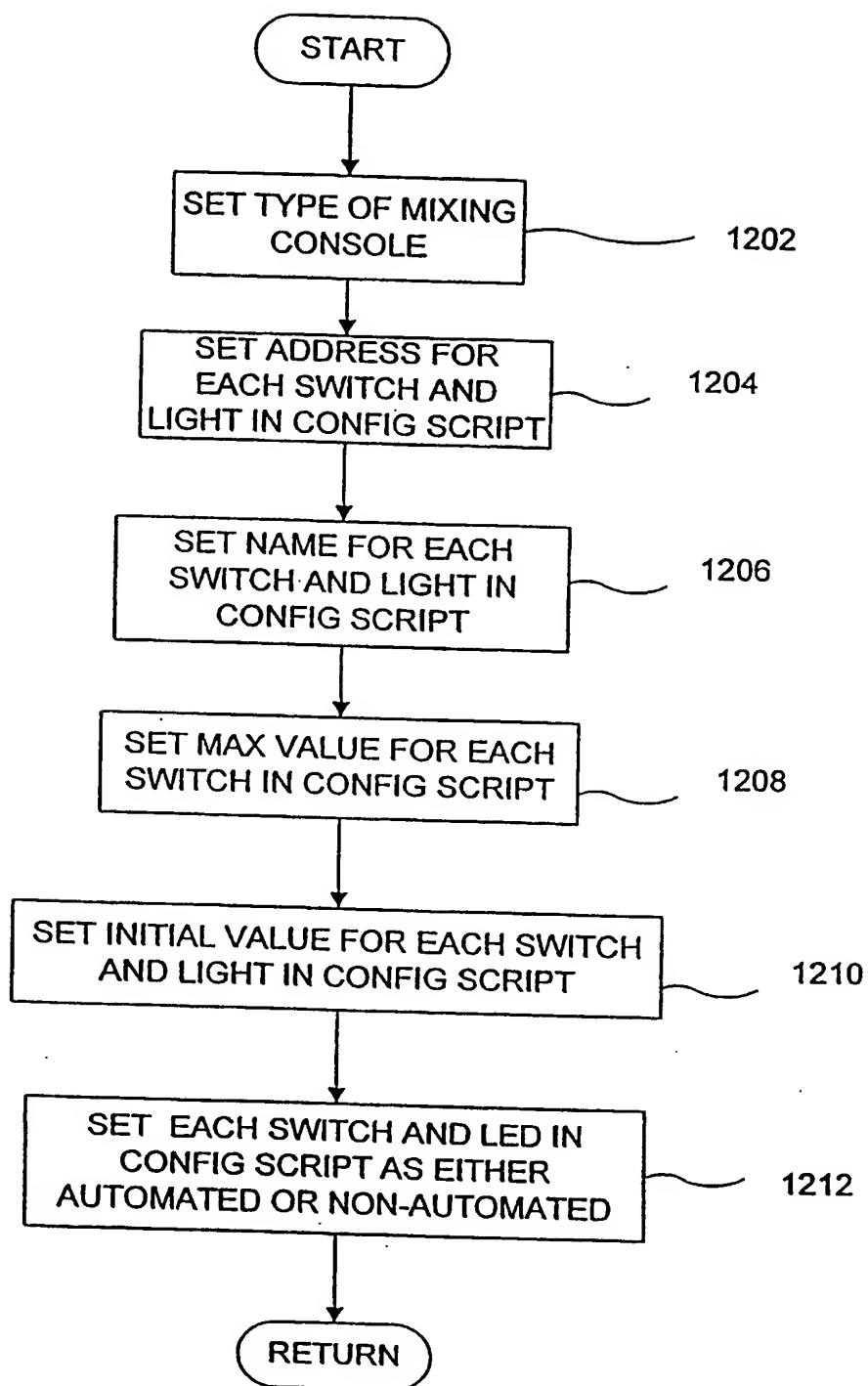
10/14

**FIG. 10**

11/14

**FIG. 11**

12/14

**FIG. 12**

13/14

FIG. 13A

type: window  
 rect: -1024,0,0,768  
 option: updateeraseoff  
 option: noclosebox  
 option: noframe  
 action: setFatControls  
 forecolor: white  
 backcolor: black  
 option: hasmenubar  
 option: hashelpbar

type: menubar  
 name: windowbar  
 rect: 750,0,948,20  
 textfont: espy  
 textjustify: right  
 newmenu: SessionMenu  
 newmenu: FaderMenu  
 option: showclock

type: image  
 rowwidth: 38  
 location: 0,0  
 imagename: 1006  
 arraysize: 24  
 textjustify: left

type: image  
 location: 912,392  
 imagename: 1113

type: image  
 location: 912,0  
 imagename: 2019

type: text  
 name: VPOTLabel  
 rect: 918,484,978,494  
 forecolor: white  
 backcolor: black  
 textfont: espy  
 textjustify: center  
 TextTransparent: true

type: fill  
 forecolor: black  
 rect: 912,241,1024,392  
 TextTransparent: false

type: button  
 rect: 948,0,1024,22  
 title: CloseAll  
 action: closeAllWindows  
 lineheight: 22  
 forecolor: black  
 textjustify: left

type: button  
 name: files  
 rect: nextline  
 title: Files  
 action: openDesktop

FIG. 13C

forecolor: black  
 option: momentary

type: button  
 rect: nextline  
 title: Surround  
 action: toggleShowSurround  
 forecolor: black  
 option: momentary

type: text  
 rect: 915,335,1024,347  
 title: Channel  
 forecolor: white  
 backcolor: black  
 textfont: espy  
 textjustify: center

type: image  
 location: 952,300  
 imagename: 2007

type: digitclick  
 name: fatchan  
 location: 955,303  
 status: transportstatus  
 address: FatChanNumber  
 option: nofocus  
 option: UseStandardImages

part file: channel.txt  
 part file: vstrip.txt

FIG. 13B

type: button  
 name: setup  
 rect: nextline  
 title: Setup  
 action: setup  
 lineheight: 22  
 option: momentary

type: button  
 name: events  
 rect: nextline  
 title: EventEditor  
 action: openGraphicList  
 option: momentary

type: button  
 name: scenes  
 rect: nextline  
 title: SceneList  
 action: openSceneList  
 option: momentary

type: button  
 name: locator.txt  
 rect: nextline  
 title: Locator  
 action: openLocator  
 option: momentary

type: button  
 name: effectA  
 rect: nextline  
 titlestart: MackieFX  
 action: viewEffectA  
 option: momentary

type: button  
 name: effectB  
 rect: nextline  
 nexttitle: MackieFX  
 action: viewEffectB  
 option: momentary

type: button  
 name: effectC  
 rect: nextline  
 nexttitle: MackieFX  
 action: viewEffectC  
 option: momentary

type: button  
 name: effectD  
 rect: nextline  
 nexttitle: MackieFX  
 action: viewEffectD  
 option: momentary

type: button  
 rect: nextline  
 title: Dynamics  
 action: toggleShowDynamics

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US98/16590

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04B 1/00

US CL :364/400.01; 381/119

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/400.01; 381/119

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

None

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

None

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 4,993,073 A (SPARKES) 12 February 1991 (12.02.91), see the entire document.	25-74
Y	US 5,442,789 A (BAKER ET AL.) 15 August 1995 (15.08.95), see the entire document.	25-74



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	*T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be of particular relevance	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A*	document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means		
*P* document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

14 OCTOBER 1998

Date of mailing of the international search report

19 NOV 1998

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

F. W. Isen

*Diane Smith for*

Telephone No. (703) 305-4386

Form PCT/ISA/210 (second sheet)(July 1992)\*